



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 05108823  
(43)Date of publication of application: 30.04.1993

(51)Int.Cl.

G06F 15/70

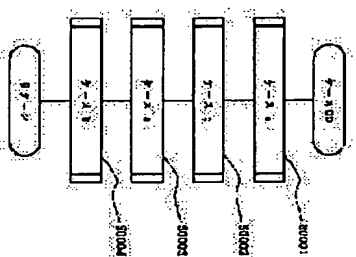
(21)Application number: 03272701 (71)Applicant: CANON INC  
(22)Date of filing: 21.10.1991 (72)Inventor: ISHIDA YOSHIHIRO

(54) CONTOUR EXTRACTING METHOD AND DEVICE THEREFOR

(57)Abstract:

**PURPOSE:** To provide the extracting method of two contour points set between two picture elements combined in an oblique direction, based on a rule at the time of extracting the contour points of an 8-combined picture element area.

**CONSTITUTION:** This method is provided with the process for setting merely one point each for constituting a contour line between a noticed picture element and a neighboring picture element at each noticed picture element, based on the state of the neighboring element, and deciding the connecting direction of the points constituting the contour line, and the process for judging the connecting state of the point constituting the contour line with another point constituting the contour line. The position of the noticed picture element is set on picture data in the order of a luster scanning, and the process is executed at each noticed picture element based on the state of the neighboring picture element, so that the contour point can be



extracted. The picture element in the neighborhood of the position of a picture element (e) is processed (step S0001). The picture element in the neighborhood of the position of a picture element C is processed (step S0002). The picture element in the neighborhood of the position of a picture element A is processed (step S0003). The picture element in the neighborhood of the position of a picture element B is processed (step S0004). Then, the processing is returned to a case 00.

LEGAL STATUS

[Date of request for examination]  
[Date of sending the examiner's decision of rejection]  
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]  
[Date of final disposal for application]  
[Patent number]  
[Date of registration]  
[Number of appeal against examiner's decision of rejection]  
[Date of requesting appeal against examiner's decision of rejection]  
[Date of extinction of right]

Copyright (C) 1998 Japanese Patent Office

(51) Int. Cl.<sup>6</sup> 識別記号 庁内整理番号 F I 技術表示箇所  
G 0 6 F 1 5 / 7 0 3 3 5 Z 4 0 7 1 - 5 L

審査請求 未請求 請求項の数 5

(全 13 頁)

(11) 出願番号 特願平 5-272701  
(12) 出願日 平成5年(1993)10月11日

(11) 出願人 000001007  
キヤノン株式会社  
東京都大田区下丸子1丁目10番2号  
(12) 発明者 石田 良弘  
東京都大田区下丸子1丁目10番2号キヤノン株式会社内  
(14) 代理人 弁理士 丸島 健一

(41) 【発明の名称】 輪郭抽出方法及び装置

(11) 【要約】 (修正有)

【目的】 8 連結の画素領域の輪郭点を抽出する場合の規則につき、斜め方向に連結する 2 つの画素間に設定する 2 つの輪郭点の、抽出法を提供する。

【構成】 注目画素毎に近傍画素の状態に基づいて、該注目画素と該近傍画素間に輪郭線を構成する点をそれぞれ高々一点まで設定し、該輪郭線を構成する点の接続方向を決定する工程と、前期輪郭線を構成する点と輪郭線を構成する他の点との接続状態を判断する工程とを有し、前期注目画素の位置をラスタ走査順に画像データ上に設定して、注目画素毎に近傍画素の状態に基づいて上記工程を実行し、輪郭点を抽出することができ、輪郭抽出方法である。画素 e の位置の画素近傍の処理を行うステップ S0001。画素 c の位置の画素近傍の処理を行うステップ S0002。A の位置の画素近傍の処理を行うステップ S0003。B の位置の画素近傍の処理を行うステップ S0004。これからケース 00 へ戻る。



【特許請求の範囲】

【請求項 1】 注目画素毎に近傍画素の状態に基づいて、該注目画素と該近傍画素間に輪郭線を構成する点と、該注目画素と高々一点まで設定し、該輪郭線を構成する点の接続方向を決定する工程と、前期輪郭線を構成する点と輪郭線を構成する他の点との接続状態を判断する工程とを有し、前期注目画素の位置をラスタ走査順に画像データ上に設定して、注目画素毎に近傍画素の状態に基づいて上記工程を実行し、輪郭点を抽出することを特徴とする画素の輪郭抽出方法。

【請求項 2】 前期輪郭線を構成する点の位置が入力画素間の中間位置であることを特徴とする請求項 1 に記載の輪郭抽出方法。

【請求項 3】 抽出された輪郭点の位置が入力画素間の位置であることを特徴とする請求項 1 に記載の輪郭抽出方法。

【請求項 4】 画像データにおける注目画素と、その近傍画素の状態を保持する保持手段と、前記注目画素及びその近傍画素をラスタ走査順に更新する手順と、前記注目画素とその近傍画素の状態に基づいて、該注目画素と該近傍画素間のそれぞれに水平方向、または垂直方向の輪郭ベクトルを高々一つ抽出する検出手段と、前記輪郭ベクトル同士との接続状態を判断する判断手段と、前記判断手段により判断された接続状態をもとに、前記画像データ上の輪郭を抽出する抽出手段を有することを特徴とする輪郭抽出装置。

【請求項 5】 前記近傍画素は、前記注目画素を取り囲む 8 個の近傍画素であることを特徴とする請求項 4 に記載の輪郭抽出装置。

【発明の詳細な説明】

【0001】 (産業上の利用分野) 本発明は、ラスタ走査順に記憶された 2 値画像の輪郭線を抽出する方法及び装置に関するものである。

【0002】

【従来の技術】 この種の装置に関して、出願人は既に特開平 2-281958 (以下先願特許と称す) を出願している。この先願特許において、『注目画素と、その近傍画素の状態により、予め定められた位置を輪郭線を構成する点とし、近傍画素の状態により輪郭線の点の接続方向を決定する工程と、前期輪郭線を構成する点と、輪郭線の他の点との接続状態を判断する工程と、前記注目画素の位置をラスタ走査順に画像データを更新し、注目画素毎に近傍画素の状態に基づいて前記工程を実行し、輪郭点を抽出する。』

【0003】 以上の構成において、画像データにおける注目画素と、その近傍画素の状態を保持し、この注目画素をラスタ走査順に取り出し、その注目画素とその近傍画素の状態に基づいて、水平方向及び垂直方向の画素配列ベクトルを抽出する。これら画素配列ベクトル同士の

接続状態を判別して、この判別された画素配列ベクトルの接続状態をもとに、画像データの輪郭を抽出するように動作する。』という手法を提案している。該手法は、画像の中の全ての輪郭線を 1 回のラスタ走査順だけで抽出でき、かつ、全画像データを記憶するだけの画像メモリを必要としないため、メモリの容量を少なくできる効果を有している。また、入力画像の画素の中心位置ではなく、画素の線単位に輪郭を抽出することによって、1 画素巾の細線に対してもある幅を有する輪郭線抽出法である。更に、原画の中の画素の 4 方向に連結した連結画素領域の輪郭線を抽出するのみならず、8 方向に連結した画素領域も抽出可能であることが紹介されている。

【0004】

【発明が解決しようとしている課題】 しかしながら、上記先願において紹介されている実施例で、8 方向に連結した (以下 “8 連結” と称する) 連結画素領域の輪郭線を抽出する場合の説明で開示される手法は、注目画素に対して、斜め方向に接続する画素間の輪郭線の抽出が複雑なものとなっていた。

【0005】 即ち、上記実施例では、注目画素が黒画素の時、注目画素の周囲 8 画素をそれぞれ白画素が黒画素かの状態を判定し、それぞれの状態のバツン毎に処理内容を決定している。注目画素の斜め方向及び斜め方向の連結画素領域の輪郭線の抽出を行う場合には、周囲画素が白画素の時のみ、輪郭点の定義をすればよいのに、斜め方向の連結画素領域の輪郭線の抽出を行う場合には、注目画素と斜め方向の画素が双方とも黒画素である場合にも輪郭点の定義を行う必要がある。このことから、斜め方向に連結画素領域の輪郭線の抽出の際には、該斜め方向に連結した 2 画素間に存在する画素間の位置は、その 2 画素のそれぞれが、注目画素位置となつた時の双方において時味され 2 回時味が行われることになる。図 2 に一例を示す。同図において、A 及び B は斜め方向に連結する 2 画素を表しており、C は輪郭点の存在する画素間位置である。このため、それぞれの時点での時味に冗長性を省き、かつ、整合性をとる必要がある。前記、

先願例においては、この斜め接続の画素領域の輪郭点の抽出手順が、その他の周囲画素の状態と独立して検出できる様に設定されておらず、各部分の抽出ルーチンをモジュール化しやすく設定されていた。図 2 の場合を例にとると、図 3 に示す様に、黒画素 A が注目画素となる状態の時、C は流出先も流入元も未定の輪郭ベクトルとして定義する。一方、黒画素 B が注目画素となる状態の時、図 4 及び図 5 に示す様に、D で示す画素位置が白画素 (図 4) か、黒画素 (図 5) かによって、処理の内容が異なる。これらの場合は、新たな輪郭ベクトルは定義せず、図 3 で定められた輪郭ベクトルの接続状態の更新のみを行っている。D で示す画素位置が白画素の

場合には、 $C_i$ の流入元と $C_j$ の流出先を決定している。  
Dで示す画面位置が黒画面の場合には、 $C_i$ の流入元のみの決定を行っている。といった具合にである。このため、プログラミンクの複雑を生み、かつ、処理速度が遅くならないという点で不適切であった。また、ハード化時の回路規模の増大を招く要因となっていた。

【0006】

【問題を解決するための手段】 本発明は、上記先願例における事情を鑑みてなされたものであり、上記先願例における8連結の画面領域の輪郭点を抽出する場合の規則について、斜め方向に連結している画面領域の抽出時に、当該斜め方向に連結する2つの黒画面間に設定する2つの輪郭点を、該2つの黒画面が注目画面となるそれぞれの時点て一点づつ定数する手法をとることによって、その他の周囲画面の状態で増立して動作できる様に、各部分の抽出ルールのモジュール化を容易にしたものである。

【0007】

【発明】 前記 先願例の他の実施例におけるケース00、ケース01、ケース02、ケース03、ケース04、ケース06、ケース08、ケース09、ケース012を以下の方式で処理する様に、以下、それぞれの処理を図6〜図26に示すフローチャートに従って説明する。

【0008】 なお、上記ケースは、先願特許の8連結の画面領域の輪郭点抽出時において、斜め方向に連結している画面領域の存在の可能性を有するそれぞれの場合に、対応しており、以上の各ケースを全て考慮すること、斜め方向に連結する全ての場合を網羅している。  
【0009】 本発明は、基本的に上述の先願特許で記述した手法での8連結領域に関する輪郭点の抽出方法を、改良することにより一段と効率的な輪郭点を抽出することが主目的であるため、実施例も上述先願特許で記述したものを元に、その改良部分のみをここに表記する形式で説明しゆく。また、8連結領域に関する輪郭点の抽出部分以外に関しては、先願特許で記述した手法をそのまま踏襲した形で実施される。そのため、新たに付した図面以外は、上記先願特許内で用いた図面をそのまま参照する形で説明する。

【0010】 図6は、先願特許の第60図で示される場合(ケース00)を処理するフローチャートである。ステップS0001では、先願特許の第60図の画面がトリクス601の

【0011】

【外1】

⑨  
の位置の画面近傍の処理を行い、その内容は図7のフローチャートで示される。この

【0012】

【外2】

⑩  
の位置は、先願特許の図1における画面Dの位置に対応している。

【0013】 図7のフローチャートにおいて、ステップS2001では、 $(2i-1, 2j-1)$ を水平ベクトルの地点として、先願特許の第23図で示される水平ベクトルの登録テーブルに登録する。ステップS2002では、この水平ベクトルの流出先のベクトルが未定であるとして、先願特許の第32図で示されるテーブル32.1に登録し、カウンタ32.0を更新する。ステップS2003では、前記画面Dの位置が黒画面( $f(D)=1$ )か否( $f(D)=0$ )かを判定し、黒画面、即ち $f(D)=1$ の場合は、ステップS2006へ進み、そうでない場合は、即ち白画面 $f(D)=0$ の場合は、ステップS2004へ進む。ステップS2004では、前記水平ベクトルの流入元ベクトルもまた未定であるとして、先願特許の第31図で示されるテーブル31.1に登録し、カウンタ31.0を更新する。ステップS2005では、水平ベクトルカウンタ23.0を+1して、元の処理に戻る。また、ステップS2006では、ステップS2001で定義した水平ベクトルに流入する垂直ベクトルをサーチする。そのサーチの手順は、先願特許の第30図で説明されている手法に準じる。ただし同図のステップS52で「項目番号( $k-1$ )」が示す垂直ベクトルの地点のx座標が $(2i+1)$ か?とある所を、 $f(2i-1)$ 〜 $j$ として動作するものである。ステップS2006の処理を終えると、ステップS2005へ進む。かくして、図6ステップS0001のケース00の処理を終える。ここでは、図8及び図9で示される様に、注目画面と画面位置Dとの間にある水平ベクトルの一つの抽出を行っているものである。(ここで、0は水平ベクトルの地点を表し、0印から出る矢印は、水平ベクトルの向く方向(即ち、左または右)を表し、0印に入る矢印は、該水平ベクトルの地点に注入してくる垂直ベクトルの向く方向(即ち、上または下)を表している。また、破線矢印は、接続関係が未定であることを意味し、実線矢印は接続関係がその時点で確定するベクトルであることを意味している。

【0014】 次に、ステップS0002へ進み、先願特許の第1図に示すCの位置の画面近傍の処理を行う。その内容は、図10のフローチャートで示される。ステップS2101では、 $(2i-1, 2j+1)$ を垂直ベクトルの地点として、先願特許の第24図で示される垂直ベクトルの登録テーブルに登録する。ステップS2102では、この垂直ベクトルの流入元のベクトルが未定であるとして、先願特許の第33図で示されるテーブル33.1に登録し、カウンタ33.0を更新する。ステップS2103では、前記画面Cの位置が黒画面( $f(C)=1$ )か否( $f(C)=0$ )かを判定し、黒画面、即ち、 $f$

50

(C)=1の場合は、ステップS2104へ進み、そ

でない場合、即ち白画面 $f(C)=0$ の場合は、ステップS2106へ進む。ステップS2104では、前記垂直ベクトルの流出先のベクトルも未定であるとして、先願特許の第34図で示されるテーブル34.1に登録し、カウンタ34.0を更新する。ステップS2105では、垂直ベクトルカウンタ24.0を+1して元の処理に戻る。また、ステップS2106では、ステップS2104で定義した垂直ベクトルが流出する水平ベクトルをサーチする。そのサーチの手順は、先願特許の第35図で説明されている手法に準じる。ステップS2106の処理を終えると、ステップS2005へ進む。かくして、図6ステップS0002のケース00の処理を終える。ここでは、図11及び図12で示される様に、注目画面と画面位置Cとの間にある垂直ベクトルの一つの抽出を行っているものである。(ここで、△印は、垂直ベクトルの地点を表し、△印から出る矢印は、垂直ベクトルの向く方向(即ち、上または下)を表し、△印に入る矢印は、該垂直ベクトルの地点に流入してくる水平ベクトルの向く方向(即ち、左または右)を表している。また、破線矢印は、接続関係が未定であることを意味し、実線矢印は、接続関係がその時点で確定するベクトルであることを意味している。

【0015】 次に、ステップS0003へ進み、先願特許の第1図に示すAの位置の画面近傍の処理を行う。その内容は、図13のフローチャートで示される。ステップS2201では、 $(2i+1, 2j-1)$ を垂直ベクトルの地点として、先願特許の第24図で示される垂直ベクトルの登録テーブルに登録する。ステップS2202では、ステップS2201で定義した垂直ベクトルに流入してくる水平ベクトルをサーチする。そのサーチの手順は、先願特許の第45図で説明されている手法に準じる。ステップS2203では、前記画面Aの位置が黒画面( $f(A)=1$ )か否( $f(A)=0$ )かを判定し、黒画面、即ち、 $f(A)=1$ の場合は、ステップS2206へ進み、そうでない場合、即ち、白画面 $f(A)=0$ の場合は、ステップS2204へ進む。ステップS2204では、前記垂直ベクトルの流出先のベクトルは未定であるとして、先願特許の第34図で示されるテーブル34.1に登録し、カウンタ34.0を更新する。ステップS2205では、垂直ベクトルカウンタ24.0を+1して元の処理に戻る。また、ステップS2206では、ステップS2201で定義した垂直ベクトルが流出する水平ベクトルをサーチする。そのサーチの手順は、先願特許の第35図で説明されている手法に準じる。ただし、同図のステップS62で「項目番号( $k-1$ )」が示すテーブル31.1の水平ベクトルの地点のx座標が $(2i-1)$ か?とある所を、 $f(2i+1)$ 〜 $j$ として動作するものである。ステップS2206の処理を終えると、ステップS2205へ進む。かくして、図6ステップS0003のケース00の処理を終える。ここでは、図14及び図15で

示される様に、注目画面と画面位置Aとの間にある垂直ベクトルの一つの抽出を行っているものである。  
【0016】 次に、ステップS0004に進み、先願特許の第1図に示すBの位置の画面近傍の処理を行う。その内容は、図16のフローチャートで示される。ステップS2301では、 $(2i+1, 2j+1)$ を水平ベクトルの地点として、先願特許の第23図で示される水平ベクトルの登録テーブルに登録する。ステップS2302では、ステップS2301で定義した水平ベクトルが流出してくる垂直ベクトルをサーチする。そのサーチの手順は、先願特許の第46図で説明されている手法に準じる。ステップS2303では、前記画面Bの位置が黒画面( $f(B)=1$ )か否( $f(B)=0$ )かを判定し、黒画面、即ち $f(B)=1$ の場合は、ステップS2304へ進み、そうでない場合、即ち白画面 $f(B)=0$ の場合は、ステップS2306へ進む。ステップS2304では、前記水平ベクトルの流入元のベクトルは未定であるとして、先願特許の第31図で示されるテーブル31.1に登録し、カウンタ31.0を更新する。ステップS2305では、水平ベクトルカウンタ23.0を+1して元の処理に戻る。また、ステップS2306では、ステップS2301で定義した水平ベクトルの流入元の垂直ベクトルをサーチする。そのサーチの手順は、先願特許の第30図で説明されている手法に準じる。ステップS2306の処理を終えると、ステップS2305へ進む。かくして、図6ステップS0004のケース00の処理を終える。これで、先願特許の第60図で示される場合(ケース00)の処理を終了する。  
【0017】 図19は、先願特許の第61図で示される場合(ケース01)を処理するフローチャートである。  
【0018】 まず、ステップS0101では、先願特許の第61図の

【外3】

【外3】

⑪

の位置の近傍の処理を行う。その処理内容は、前述の図10に示したケース0と全く同様である。次にステップS0102に進み、先願特許の第61図の

【外4】

⑫

の位置の近傍の処理を行う。その処理内容は、前述の図16に示したケース0と全く同様である。かくして、先願特許の第61図で示される場合(ケース01)の処理を終了する。

【0021】 図20は、先願特許の第62図で示される場合(ケース02)を処理するフローチャートである。  
【0022】 まず、ステップS0201では、先願特許の第62図の

【外5】

【外5】

50

の位置の近傍の処理を行う。その処理内容は、前述の図10に示したケース0と全く同様である。次にステップS0102に進み、先願特許の第61図の



画面の画面位置 (i, j) を知ることができる。また、入力ポート513及び514を介して、注目画面及びその近傍の8方向の画面の状態を知ることができる。  
 【0060】このようにして注目画面の処理が終了すると、CPU519は入力側ポート517を介して9個のラッチに記憶される画面データの更新を指示し、同時に画面データの更新完了の信号をリセットする。入力側ポート517は、この更新の指示を受けると、画面データの更新指示の信号をクリアするとともに、後段のラッチにラッチされる画面データを更新し、この更新が終了すると入力側ポート517に、更新完了の信号を出力する。

【0061】CPU519は、更新指示の出力後、出力側ポート517より更新完了の信号が入力されるのを監視している。この更新完了の信号が入力されると、新たに9個のラッチに記憶された画面データに関する処理を実行し、以下同様これを繰り返すものである。また、入力側ポート517は、画面領域の最終画面を注目画面として処理し終わった際に、入力側ポート517に終了信号を出力する。

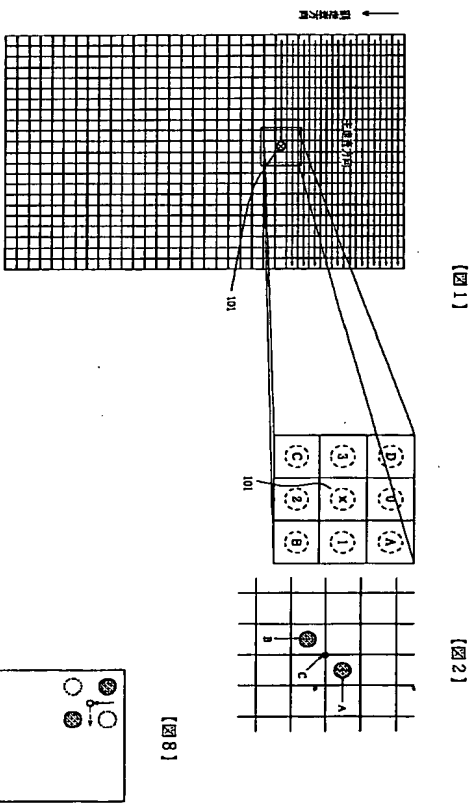
【0062】

【発明の効果】以上説明した様に、上記実施例における8方向の画面領域の端部点を抽出する場合の規則について、始めの方向に連結している2つの黒画面面に設定する2つの端部点を、該2つの黒画面面が注目画面となるそれぞれの端部点を点つづ定義する手法をとることによって、その他の周囲画面の状態と独立して動作できる様に構成することを可能とし、かつ、各部の抽出手段をモジュールとして構成しやすきことができる。これにより、ハード化時の回路構成規模の減小が得られるという効果を有する。また、ソフトウェアでプログラムするにしても、プログラムの簡明に構成できることから、処理速度の向上が得られるという効果を有する。

【図面の簡単な説明】

【図1】 画面の走査方向及び注目画面と、その近傍画面の関係を示す図。  
 【図2】 先駆特種での処理を説明するための、入力側の例を示す図。  
 【図3】 先駆特種における、図2の入力に対する処理を説明する図。  
 【図4】 先駆特種における、図2の入力に対する処理を説明する図。  
 【図5】 先駆特種における、図2の入力に対する処理を説明する図。  
 【図6】 本発明における先駆特種のケース00の改良された処理内容を示すフローチャート。  
 【図7】 ケースeの処理を示すフローチャート。

【図8】 ケースeの処理内容を示す図。  
 【図9】 ケースeの処理内容を示す図。  
 【図10】 ケースcの処理内容を示す図。  
 【図11】 ケースcの処理内容を示す図。  
 【図12】 ケースcの処理内容を示す図。  
 【図13】 ケースaの処理内容を示す図。  
 【図14】 ケースaの処理内容を示す図。  
 【図15】 ケースaの処理内容を示す図。  
 【図16】 ケースbの処理内容を示す図。  
 【図17】 ケースbの処理内容を示す図。  
 【図18】 ケースbの処理内容を示す図。  
 【図19】 本発明における先駆特種のケース01の改良された処理内容を示すフローチャート。  
 【図20】 本発明における先駆特種のケース02の改良された処理内容を示すフローチャート。  
 【図21】 本発明における先駆特種のケース03の改良された処理内容を示すフローチャート。  
 【図22】 本発明における先駆特種のケース04の改良された処理内容を示すフローチャート。  
 【図23】 本発明における先駆特種のケース06の改良された処理内容を示すフローチャート。  
 【図24】 本発明における先駆特種のケース08の改良された処理内容を示すフローチャート。  
 【図25】 本発明における先駆特種のケース09の改良された処理内容を示すフローチャート。  
 【図26】 本発明における先駆特種のケース012の改良された処理内容を示すフローチャート。  
 【図27】 それぞれケース00、ケース03、ケース06、ケース09、ケース012に対する他の実施例を示すフローチャート。  
 【図28】 それぞれケース00、ケース03、ケース06、ケース09、ケース012に対する他の実施例を示すフローチャート。  
 【図29】 それぞれケース00、ケース03、ケース06、ケース09、ケース012に対する他の実施例を示すフローチャート。  
 【図30】 それぞれケース00、ケース03、ケース06、ケース09、ケース012に対する他の実施例を示すフローチャート。  
 【図31】 それぞれケース00、ケース03、ケース06、ケース09、ケース012に対する他の実施例を示すフローチャート。  
 【図32】 ハード構成を示す図。  
 【符号の説明】  
 519 CPU  
 520 メモリ

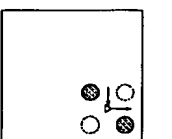
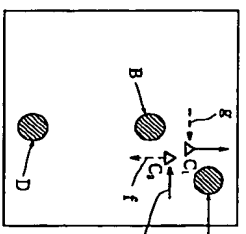
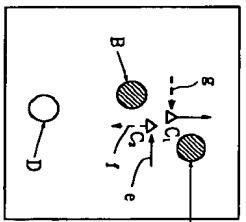
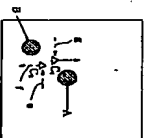


【図3】

【図4】

【図5】

【図14】



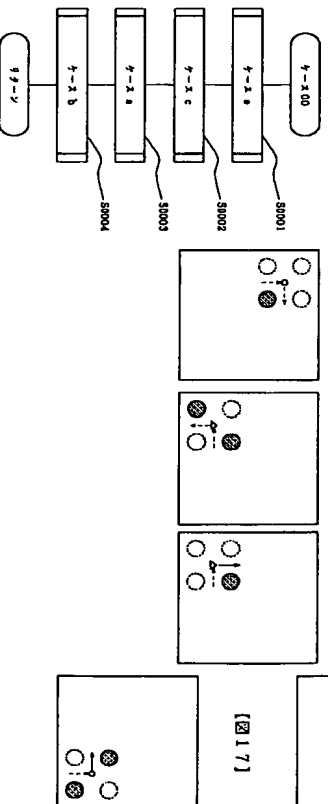
【図6】

【図9】

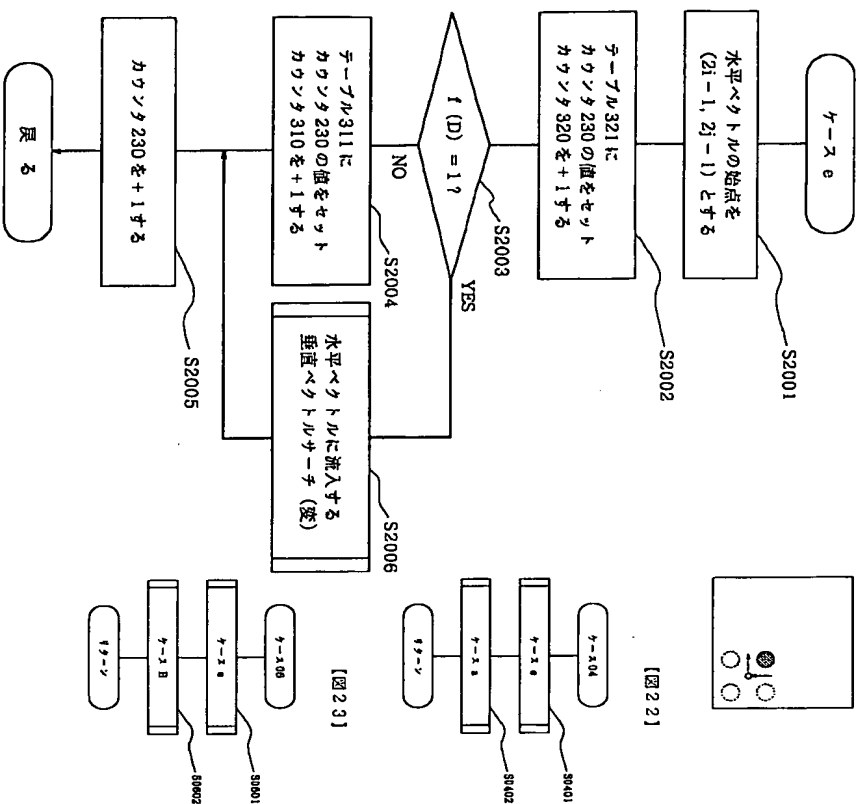
【図11】

【図12】

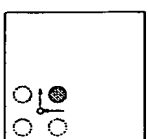
40



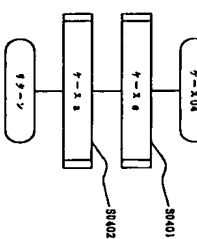
【図7】



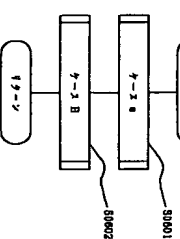
【図18】



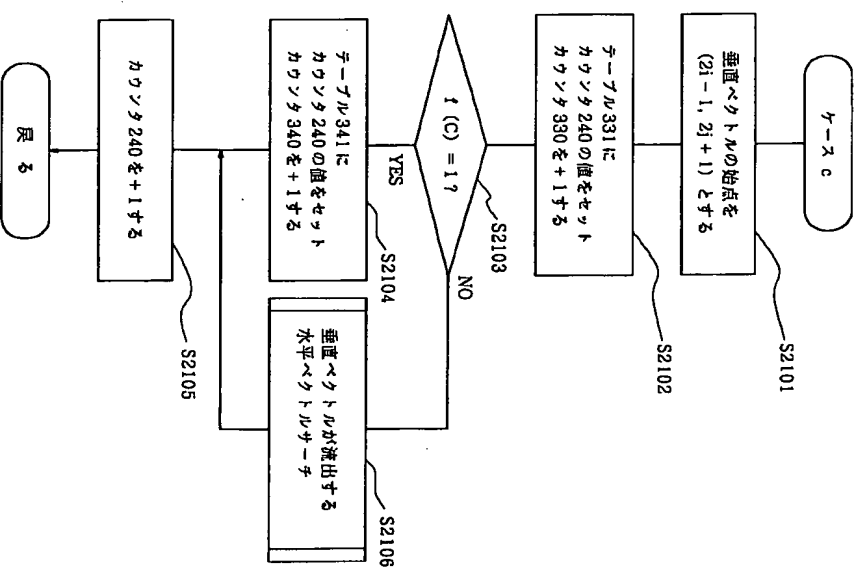
【図22】



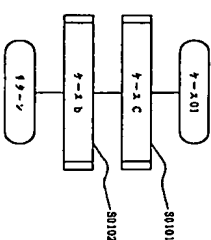
【図23】



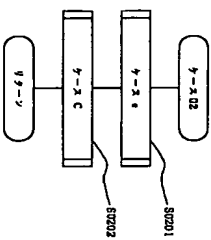
【図10】



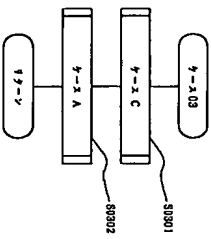
【図19】



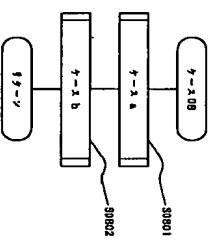
【図20】



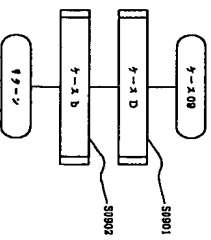
【図21】



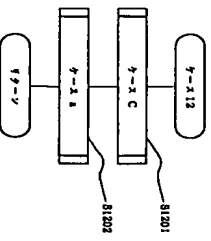
【図24】



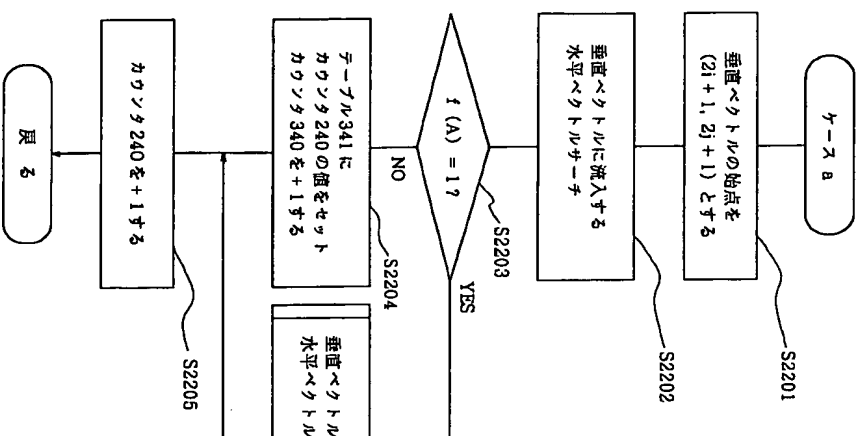
【図25】



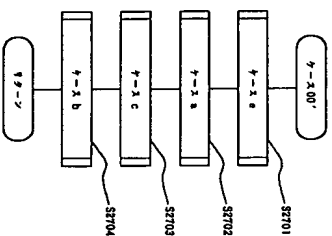
【図26】



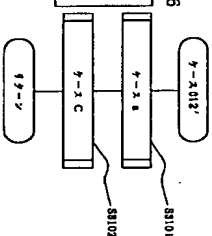
【图 13】



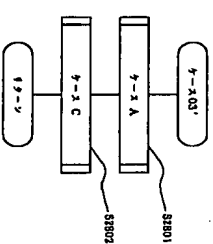
【☑27】



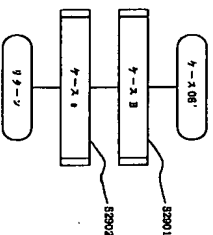
【31】



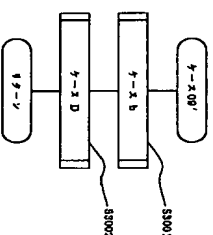
【例28】



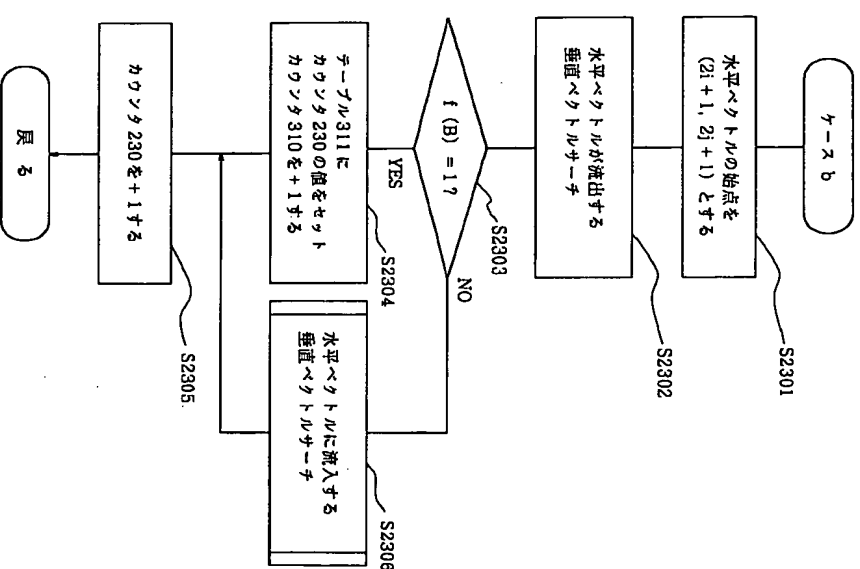
【図29】

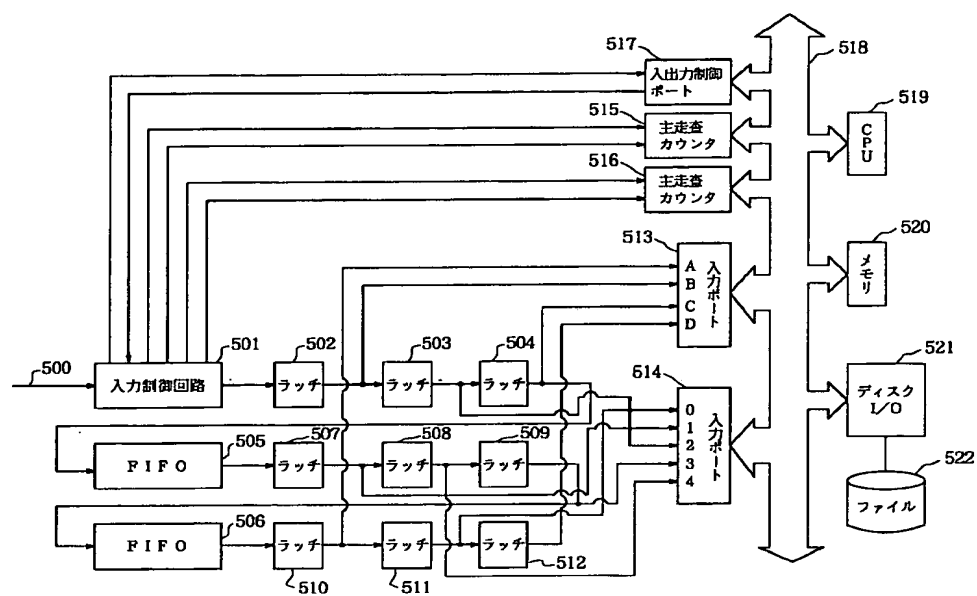


【30】



【916】





【図 32】

(13)

特開平 5-108823



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**